

CLAIMS

1. (Previously Presented) An apparatus for smoothly playing a pre-determined sequence of songs transmitted from a server over the Internet, the apparatus comprising a personal computer having a processor, a first memory that stores at least one program used by said processor to control the playing of the sequence of songs, and a second memory which is available to said at least one program for operations,

wherein said at least one program causes said processor at least to:

as soon as a song starts to play, start to download, consecutively, a first small portion of each of a number of songs which are, in the pre-determined sequence, subsequent to the song playing in an alternating fashion, said downloaded small portions being pre-cached in a pre-cache buffer which is an area in said second memory;

as soon as the user skips to a target song whose first small portion has been pre-cached, start to play the first small portion of said target song,

at the same time as said target song starts to play, deleting all pre-cached songs preceding said target song in said pre-determined sequence ; and

at the same time start to download the rest of said target song so that as soon as the playing of the first small portion of said target song ends, start to play the rest of said target song which is being downloaded from the server over the Internet,

wherein playtime of said downloaded first small portion is limited to comply with royalty requirements.

2. (Original) The apparatus of Claim 1, wherein said first small portion is approximately the data required for playing the first ten seconds.

3. (Original) The apparatus of Claim 1, wherein said number is five.
4. (Original) The apparatus of Claim 1, wherein said number of songs is all songs subsequent to the song in playing.
5. (Original) The apparatus of Claim 1, wherein said buffer follows a first-in first-out algorithm and allows writing while reading.
6. (Previously Presented) A method for smoothly playing a pre-determined sequence of songs transmitted from a remote server to a local computer having a processor over the Internet, comprising the steps of:
 - (a) as soon as a song starts to play, downloading to said computer, consecutively, a first small portion of each of a number of songs which are, in the pre-determined sequence, subsequent to said song in playing in an alternating fashion,

wherein said first small portion of each of said number of songs downloaded is limited to a playtime not incurring a royalty,

wherein said number of songs is maintained at a single positive integer;
 - (b) pre-caching said downloaded small portions in a pre-cache buffer which is an area of said local device's memory;
 - (c) as soon as the user skips from a song in play to a target song, checking whether a file for said target song exists in said buffer, wherein if the check result is yes, continuing on step (d);
 - (d) playing the first small portion of said target song; and
 - (e) at the same time step (d) starts, deleting all pre-cached songs preceding said target song in said pre-determined sequence.

7. (Previously Presented) The method of Claim 6, further comprising the steps of:

(f) as soon as step (d) starts, downloading the rest of said target song; and

(g) playing the rest of said target song which is being downloaded from the server over the Internet.

8. (Previously Presented) The method of Claim 7, further comprising the step of:

(h) as soon as step (d) starts, continuing on to step (a), wherein if one or more songs subsequent to said target song are already pre-cached, skipping said one or more songs and downloading the subsequent ones, consecutively to make up said number.

9. (Original) The method of Claim 8, further comprising the steps of:

(i) if no skip command is given by the user while said target song is playing, as soon as the playing of said target song ends, playing the next song immediately subsequent to said target song; and

(j) if a skip command is given by the user while said target song is playing, continuing on step (c).

10. (previously presented) The method of Claim 7, wherein if the check result of step (c) is no, further comprising the steps of:

(k) sending request to stop transmitting of said song in playing and start transmitting said target song;

(l) deleting the pre-cached portion for any song which precedes said target song in the pre-determined sequence of songs after elapse of a predetermined, configurable time interval;

(m) downloading said target song;

(n) playing said target song while being downloaded as soon as said buffer allows so; and

(o) at the same time with step (n), continuing on step (a).

11. (Previously Presented) The method of Claim 10, subsequent to step (n), further comprising the steps of:

(p) if another skip command is given by the user while said target song is playing, continuing on to step (c); and

(q) if no skip command is given by the user while said target song is playing, as soon as the playing of said target song ends, playing the first small portion of the next song subsequent to said target song;

(r) at the same time with step (q), downloading the rest of said target song;

(s) at the same time with step (q), continuing on step (a), wherein if one or more songs subsequent to said next song are already pre-cached, skipping said one or more songs and downloading the subsequent ones, consecutively, to make up said number; and

(t) subsequent to step (q), playing the rest of the next song which is being download from the server over the Internet

12. (Original) The method of Claim 6, wherein said first small portion is approximately the data required for playing the first ten seconds.

13. (Original) The method of Claim 6, wherein said number is five.
14. (Original) The method of Claim 6, wherein said number of songs is all songs subsequent to the song in playing.
15. (Original) The method of Claim 6, wherein said buffer follows a first-in first-out algorithm and allows writing while reading.
16. (Previously Presented) A program storage medium readable by a user's computer having a processor, tangibly embodying a program of instructions executable by the computer to perform a method for smoothly playing a pre-determined sequence of songs transmitted from a remote server to a local device over the Internet, comprising the steps of:
- (a) as soon as a song starts to play, downloading to said computer, consecutively, a first small portion of each of a number of songs which are, in the pre-determined sequence, subsequent to said song in playing in an alternating fashion,
 - wherein said first small portion of each of said number of songs downloaded is limited to a playtime not incurring a royalty;
 - (b) pre-caching said downloaded small portions in a pre-cache buffer which is an area of said local device's memory
 - (c) as soon as the user skips from a song in playing to a target song, checking whether a file for said target song exists in said buffer, wherein if the check result is yes, continuing on to step (d);
 - (d) playing the first small portion of said target song; and
 - (e) at the same time as step (d) starts, deleting all pre-cached songs preceding said target song in said pre-determined sequence.

17. (Previously Presented) The program storage medium of Claim 16, further comprising the steps of:

(f) as soon as step (d) starts, downloading the rest of said target song;

and

(g) playing the rest of said target song which is being downloaded from the server over the Internet.

18. (Previously Presented) The program storage medium of Claim 17, further comprising the step of:

(h) as soon as step (d) starts, continuing on to step (a), wherein if one or more songs subsequent to said target song are already pre-cached, skipping said one or more songs and downloading the subsequent ones, consecutively to make up said number.

19. (Previously Presented) The program storage medium of Claim 18, further comprising the steps of:

(i) if no skip command is given by the user while said target song is playing, as soon as the playing of said target song ends, playing the next song immediately subsequent to said target song; and

(j) if a skip command is given by the user while said target song is playing, continuing on to step (c).

20. (Original) The program storage medium of Claim 17, wherein if the check result of step (c) is no, further comprising the steps of:

(k) sending request to stop transmitting of said song in playing and start transmitting said target song;

(l), at the same time with step (k), deleting the pre-cached portion for any song which is prior to said target song in the pre-determined sequence of songs;

(m) downloading said target song;

(n) playing said target song while being downloaded as soon as said buffer allows so; and

(o) at the same time with step (n), continuing on step (a).

21. (Previously Presented) The program storage medium of Claim 20, subsequent to step (n), further comprising the steps of:

(p) if another skip command is given by the user while said target song is playing, continuing on to step (c); and

(q) if no skip command is given by the user while said target song is playing, as soon as the playing of said target song ends, playing the first small portion of the next song subsequent to said target song;

(r) at the same time with step (q), downloading the rest of said target song;

(s) at the same time with step (q), continuing on to step (a), wherein if one or more songs subsequent to said next song are already pre-cached, skipping said one or more songs and downloading the subsequent ones, consecutively to make up said number; and

(t) subsequent to step (q), playing the rest of the next song which is being download from the server over the Internet

22. (Original) The program storage medium of Claim 16, wherein said first small portion is approximately the data required for playing the first ten seconds.

23. (Original) The program storage medium of Claim 16, wherein said number is five.

24. (Original) The program storage medium of Claim 16, wherein said number of songs is all songs subsequent to the song in playing.

25. (Original) The program storage medium of Claim 16, wherein said buffer follows a first-in first-out algorithm and allows writing while reading.

26. (Previously Presented) The apparatus of Claim 1, wherein said number of songs that are cached is specified and wherein size of said buffer is also specified.

27. (Previously Presented) The apparatus of Claim 1, wherein said length of said first small portion and said number of songs are configurable in a function.

28. (Previously Presented) The apparatus of Claim 1, further comprising a function to close and remove all of said first small portion of each of said number of songs.

29. cancelled

30. cancelled

31. (Previously Presented) A method for smoothly playing a pre-determined sequence of videos transmitted from a remote server to a local computer having a processor over the Internet, comprising the steps of:

(a) as soon as a video starts to play, downloading to said computer, consecutively, a first small portion of each of a number of videos which are, in said pre-determined sequence, subsequent to said video in playing in an alternating fashion,

wherein said first small portion of each of said number of videos downloaded is limited to a playtime not incurring a royalty;

(b) pre-caching said downloaded small portions in a pre-cache buffer which is an area of said local device's memory;

(c) as soon as the user skips to a target video whose first small portion has been pre-cached, start to play the first small portion of said target video; and

(d) at the same time as said target video starts to play, deleting all pre-cached video preceding said target song in said pre-determined sequence.

32. (Previously Presented) The method of Claim 31, further comprising the step of:

a user controlling said pre-determined sequence of videos using a graphical user interface.